Termination Analysis of Loops

Zohar Manna

with Aaron R. Bradley

Computer Science Department

Stanford University

Example: GCD Algorithm

$$gcd(y_1, y_2) = \begin{cases} gcd(y_1 - y_2, y_2) & \text{if } y_1 > y_2 \\ gcd(y_1, y_2 - y_1) & \text{if } y_1 < y_2 \\ y_1 & \text{if } y_1 = y_2 \end{cases}$$

Example:

$$\begin{aligned} gcd(77, 112) &= gcd(77, 35) = gcd(42, 35) = gcd(7, 35) \\ &= gcd(7, 28) = gcd(7, 21) = gcd(7, 14) \\ &= gcd(7, 7) = 7 \end{aligned}$$

Example: GCD Program

$\texttt{int}$ **gcd**($\texttt{int}\ y_1 > 0,\ \texttt{int}\ y_2 > 0$)

while $y_1 \neq y_2$ **do**

if $y_1 > y_2$ **then** $y_1 := y_1 - y_2$ **else** $y_2 := y_2 - y_1$

done

return y_1

Abstract program:

$$\begin{aligned}
\Theta: &\quad \{y_1 \geq 1,\ y_2 \geq 1\} \\
\tau_1: &\quad \{y_1 \geq y_2 + 1\} \Rightarrow \{y_1' = y_1 - y_2,\ y_2' = y_2\} \\
\tau_2: &\quad \{y_2 \geq y_1 + 1\} \Rightarrow \{y_2' = y_2 - y_1,\ y_1' = y_1\}
\end{aligned}$$

for $y_1, y_2 \in \mathbb{R}$

Example: Termination of GCD

$$\begin{aligned}
\Theta: &\quad \{y_1 \geq 1,\ y_2 \geq 1\} \\
\tau_1: &\quad \{y_1 \geq y_2 + 1\} \Rightarrow \{y_1' = y_1 - y_2,\ y_2' = y_2\} \\
\tau_2: &\quad \{y_2 \geq y_1 + 1\} \Rightarrow \{y_2' = y_2 - y_1,\ y_1' = y_1\}
\end{aligned}$$

$\boxed{\delta(y_1, y_2) = y_1 + y_2}$ is a **ranking function**

$\boxed{y_1 \geq 1 \wedge y_2 \geq 1}$ is a **loop invariant**

- δ is **bounded from below**:
  if τ_1 or τ_2 can be taken, $\delta(y_1, y_2) \geq 0$
- δ **decreases** on each iteration:
  if τ_1 or τ_2 is taken, $\delta(y_1', y_2') \leq \delta(y_1, y_2) - 1$

Therefore, GCD terminates.

Goal:
Find ranking functions and supporting invariants automatically.

Ranking Functions

Loops

Loop Abstraction: GCD

$L : \langle \Theta, \mathcal{T} \rangle$ over $\mathcal{V}$:

- **variables** $\mathcal{V}$ range over $\mathbb{R}$ $\{y_1, y_2\}$
- **initial condition** Θ is assertion over $\mathcal{V}$ $y_1 \geq 1 \wedge y_2 \geq 1$
- **transitions** $\tau \in \mathcal{T}$ are assertions $\{\tau_1, \tau_2\}$
  $\tau(\mathcal{V}, \mathcal{V}')$ over $\mathcal{V} \cup \mathcal{V}'$

Loop Validity:

Assertion φ is **valid over loop** L

$$L \models \varphi$$

if φ holds on all **reachable states** $\mathcal{S}_L$ of L. *values of* (y_1, y_2)

In practice, replace "$L \models$" with **loop invariants**. $y_1 \geq 1 \wedge y_2 \geq 1$

Well-founded Relation

$(D, \prec)$: $\prec$ is **well-founded** if there is no infinite sequence

$$d_1, d_2, d_3, \ldots \quad \text{where } d_i \in D$$

such that

$$(\forall i)\ d_i \succ d_{i+1}$$

$(d_2 \prec d_1 \ \Leftrightarrow \ d_1 \succ d_2)$

Examples:

- $(\mathbb{Z}^+, <)$
- $(\mathbb{R}^+, \prec_\epsilon)$ for $\epsilon > 0 \qquad x \prec_\epsilon y \ \Leftrightarrow \ x \leq y - \epsilon$
- $(\mathbb{L}, \prec)$ for lists $\mathbb{L} \qquad \ell_1 \prec \ell_2 \ \Leftrightarrow \ |\ell_1| < |\ell_2|$

Ranking Function

Consider loop $L : \langle \Theta, \mathcal{T} \rangle$ over $\mathcal{V}$.
$\delta : \mathcal{S}_L \to \mathbb{R}$ is a **ranking function** of L if

(Bounded) $(\forall \tau \in \mathcal{T})$

$$L \models \quad \tau(\mathcal{V}, \mathcal{V}') \ \rightarrow \ \delta(\mathcal{V}) \geq 0$$

(Ranking) $(\exists \epsilon > 0)(\forall \tau \in \mathcal{T})$

$$L \models \quad \tau(\mathcal{V}, \mathcal{V}') \ \rightarrow \ \delta(\mathcal{V}') \leq \delta(\mathcal{V}) - \epsilon$$

δ, ϵ induce a well-founded relation over $\mathcal{S}_L$:
for $s, t \in \mathcal{S}_L$,

$$s \prec t \quad \Leftrightarrow \quad \delta(s) \leq \delta(t) - \epsilon$$

Thus, L always terminates.

Example: GCD

Prove $\delta(y_1, y_2) = y_1 + y_2$ is a ranking function for GCD.

- Take loop invariant $y_1 \geq 1 \wedge y_2 \geq 1$.
- Choose $\epsilon = 1$.

Bounded τ_1

$$\underbrace{y_2 \geq 1}_{\text{invariant}} \wedge \underbrace{y_1 \geq y_2 + 1}_{\text{guard of } \tau_1} \rightarrow y_1 + y_2 \geq 0$$

Ranking τ_1

$$\underbrace{y_2 \geq 1}_{\text{invariant}} \rightarrow \underbrace{(y_1 - y_2) + (y_2)}_{\text{substitution by } \tau_1} \leq y_1 + y_2 - \underbrace{1}_{\epsilon}$$

Example: GCD

Bounded τ_2

$$\underbrace{y_1 \geq 1}_{\text{invariant}} \wedge \underbrace{y_2 \geq y_1 + 1}_{\text{guard of } \tau_2} \rightarrow y_1 + y_2 \geq 0$$

Ranking τ_2

$$\underbrace{y_1 \geq 1}_{\text{invariant}} \rightarrow \underbrace{(y_1) + (y_2 - y_1)}_{\text{substitution by } \tau_2} \leq y_1 + y_2 - \underbrace{1}_{\epsilon}$$

Assertions are valid, so GCD always terminates.

Lexicographic Well-founded Relation

Given well-founded relations over domains

$$(D_1, \prec_1), (D_2, \prec_2), \ldots, (D_k, \prec_k)$$

define **lexicographic well-founded relation** $\prec$ over

$$D = D_1 \times D_2 \times \cdots \times D_k$$

For $d = \langle d_1, d_2, \ldots, d_k \rangle,\ e = \langle e_1, e_2, \ldots, e_k \rangle \in D$

$$d \prec e \quad \Leftrightarrow \quad (\exists i)\,[d_i \prec_i e_i \ \wedge\ (\forall j < i)\ d_j = e_j]$$

$$\begin{array}{ccccccc} \langle & d_1, & \ldots, & d_i, & \ldots, d_k & \rangle \\ & = & = & \prec_i & & \\ \langle & e_1, & \ldots, & e_i, & \ldots, e_k & \rangle \end{array}$$

Lexicographic Ranking Function

Consider loop $L : \langle \Theta, \mathcal{T} \rangle$.

Tuple of functions $\delta : \langle \delta_1, \delta_2, \ldots, \delta_k \rangle$ where $\delta_i : \mathcal{S}_L \to \mathbb{R}$

and map $\pi : \mathcal{T} \to \{1, \ldots, k\}$

comprise a **lexicographic ranking function** for L if

(Bounded) $(\forall \tau \in \mathcal{T})$

$$L \models \quad \tau(\mathcal{V}, \mathcal{V}') \;\rightarrow\; \delta_{\pi(\tau)}(\mathcal{V}) \geq 0$$

(Ranking) $(\exists \epsilon > 0)(\forall \tau \in \mathcal{T})$

$$L \models \quad \tau(\mathcal{V}, \mathcal{V}') \;\rightarrow\; \delta_{\pi(\tau)}(\mathcal{V}') \leq \delta_{\pi(\tau)}(\mathcal{V}) - \epsilon$$

(Nonincreasing) $(\forall \tau \in \mathcal{T})$

$$L \models \quad (\forall j < \pi(\tau))[\tau(\mathcal{V}, \mathcal{V}') \;\rightarrow\; \delta_j(\mathcal{V}') \leq \delta_j(\mathcal{V})]$$

Induced Lexicographic Well-founded Relation

δ, ϵ induce a lexicographic well-founded relation over $\mathcal{S}_L$:
for $s, t \in \mathcal{S}_L$,

$$s \prec t \quad \Leftrightarrow \quad (\exists i)\,[\delta_i(s) \leq \delta_i(t) - \epsilon \ \wedge \ (\forall j < i)\ \delta_j(s) = \delta_j(t)]$$

Thus, L always terminates.

Example: McCarthy 91

For $n \in \mathbb{Z}^+$,

$$f(n) = \begin{cases} f(f(n+11)) & \text{if } n \leq 100 \\ n - 10 & \text{if } n > 100 \end{cases}$$

For every $1 \leq n \leq 92$, $f(n) = 91$, if it terminates.

We prove termination for all $n \in \mathbb{Z}^+$.

Example:

$$\begin{aligned} f(89) &= f(f(100)) = f(f(f(111))) = f(f(101)) \\ &= f(91) = f(f(102)) = \cdots = 91 \end{aligned}$$

Example: Imperative McCarthy 91

```
int f(int x)
  int s = 1
  while true do
    if x > 100
    then  if s = 1
          then return x − 10
          else  x := x − 10
                s := s − 1
    else  x := x + 11
          s := s + 1
  done
```

Abstract program:

$$\begin{aligned}
\Theta: &\quad \{s = 1\} \\
\tau_1: &\quad \{x \geq 101,\ s \neq 1\} \Rightarrow \{x' = x - 10,\ s' = s - 1\} \\
\tau_2: &\quad \{x \leq 100\} \Rightarrow \{x' = x + 11,\ s' = s + 1\}
\end{aligned}$$

for $x, s \in \mathbb{R}$

Example: McCarthy 91

Prove

$$\langle \underbrace{10s - x + 90}_{\delta_1}, \underbrace{x}_{\delta_2} \rangle$$

$$\pi(\tau_1) = 2,\ \pi(\tau_2) = 1$$

is a lexicographic ranking function for McCarthy 91.

- Take loop invariant $s \geq 1$.
- Choose $\epsilon = 1$.

Show

$$\tau_1 \ \rightarrow\ \delta_2 \geq 0 \qquad\qquad \tau_2 \ \rightarrow\ \delta_1 \geq 0$$

$$\tau_1 \ \rightarrow\ \delta_2' \leq \delta_2 - \epsilon \qquad\qquad \tau_2 \ \rightarrow\ \delta_1' \leq \delta_1 - \epsilon$$

$$\tau_1 \ \rightarrow\ \delta_1' \leq \delta_1$$

Example: McCarthy 91

Bounded τ_1: $\pi(\tau_1) = 2$

$$\underbrace{x \geq 101}_{\text{guard of } \tau_1} \rightarrow \underbrace{x}_{\delta_2} \geq 0$$

Ranking τ_1: $\pi(\tau_1) = 2$

$$\underbrace{x \geq 101}_{\text{guard of } \tau_1} \rightarrow \underbrace{(x - 10)}_{\text{substitution into } \delta_2 \text{ by } \tau_1} \leq \underbrace{x}_{\delta_2} - \underbrace{1}_{\epsilon}$$

Nonincreasing τ_1: $1 < \pi(\tau_1) = 2$

$$\underbrace{x \geq 101}_{\text{guard of } \tau_1} \rightarrow \underbrace{10(s-1) - (x-10) + 90}_{\text{substitution into } \delta_1 \text{ by } \tau_1} \leq \underbrace{10s - x + 90}_{\delta_1}$$

Example: McCarthy 91

Bounded τ_2: $\pi(\tau_2) = 1$

$$\underbrace{s \geq 1}_{\text{invariant}} \;\wedge\; \underbrace{x \leq 100}_{\text{guard of } \tau_2} \;\rightarrow\; \underbrace{10s - x + 90}_{\delta_1} \;\geq\; 0$$

Ranking τ_2: $\pi(\tau_2) = 1$

$$\underbrace{10(s+1) - (x+11) + 90}_{\text{substitution into } \delta_1 \text{ by } \tau_2} \;\leq\; \underbrace{10s - x + 90}_{\delta_1} \;-\; \underbrace{1}_{\epsilon}$$

Assertions are valid, so McCarthy 91 always terminates.

The Theoretical Landscape

Ranking Functions

Theorem Every terminating loop has a ranking function.

But in general, expressing a ranking function requires
FOL with fixpoints, which is **incomplete**.

Therefore, termination is not necessarily semi-decidable.

In fact, we show that termination is not semi-decidable
for a simple class of loops.

Interlude: Linear Loops

Consider variables $\mathcal{V} = \{x_1, x_2, \ldots, x_m\}$.

homogenous vector:

$$\mathbf{x} = (x_1, \ldots, x_m, 1)^{\mathrm{T}}$$

linear assertion:

$$\mathbf{Ax} \geq 0$$

$$\begin{bmatrix} a_{1,1} & \cdots & a_{1,m} & a_{1,m+1} \\ & & \vdots & \\ a_{k,1} & \cdots & a_{k,m} & a_{k,m+1} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_m \\ 1 \end{bmatrix} \geq \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

$$\bigwedge_{i \in \{1,\ldots,k\}} (a_{i,1}x_1 + \cdots + a_{i,m}x_m + a_{i,m+1}) \geq 0$$

Interlude: Linear Loops

Consider variables $\mathcal{V} = \{x_1, x_2, \ldots, x_m\}$.

linear loop: $L : \langle \Theta, \mathcal{T} \rangle$ in which all assertions are linear

- let $(\mathbf{x}\mathbf{x}') = (x_1, \ldots, x_m, x'_1, \ldots, x'_m, 1)^{\mathrm{T}}$
- initial condition: $\boxed{\Theta\mathbf{x} \geq 0}$
- transitions: $\boxed{\tau_i(\mathbf{x}\mathbf{x}') \geq 0}$

Theoretical Limitation

Consider loops of form:

$$\Theta : \bigwedge_{x_i \in \overline{\mathcal{V}} \subseteq \mathcal{V}} x_i = c_i$$

$$\texttt{while } \mathbf{g}^{\mathrm{T}}\mathbf{x} \geq 0 \texttt{ do}$$

$$\mathbf{x} := \underbrace{(\mathbf{A}_1 | \mathbf{A}_2 | \cdots | \mathbf{A}_k)}_{\text{nondeterministic choice}} \mathbf{x}$$

$$\texttt{done}$$

for $\mathbf{x} \in \mathbb{R}^n$.

Restricted subset of linear loops.

Theorem Termination of such loops is not semi-decidable (not recursively enumerable).

No **complete** method.

Synthesis Problem

Identify class of loops Y
and class of ranking functions Z
such that
synthesis of ranking functions of form Z is complete for Y.

Examples:

- **Linear** ranking functions for **linear** loops.
- **Lexicographic linear** ranking functions for **linear** loops.

Recent Works

Colón & Sipma 2001

Synthesis of linear ranking functions for linear loops.

Colón & Sipma 2002

Extension to nested loops and multiple paths.

Colón, Sankaranarayanan & Sipma 2003

Constraint-based linear invariant generation.

Podelski & Rybalchenko 2004

Complete method for linear loops with one transition and no initial condition. Based on linear programming.

Bradley, Manna & Sipma 2005 Polyranking-based analysis for polynomial loops with assignment.

Bradley, Manna & Sipma 2005 Synthesis of lexicographic linear ranking functions with supporting invariants.

Synthesis of Linear Ranking Functions with Supporting Invariants

Linear Ranking Function

Constraint-based approach: templates with unknown coefficients.

Consider linear loop $L : \langle \Theta, \mathcal{T} \rangle$.
For $\mathbf{r}^{\mathrm{T}}\mathbf{x}$ a template ($\mathbf{r}_i$ are unknown coefficients),
$\mathbf{r}^{\mathrm{T}}\mathbf{x}$ is a **linear ranking function** if

(Bounded) $(\forall \tau \in \mathcal{T})$

$$L \models \quad \tau(\mathbf{x}\mathbf{x}') \geq 0 \ \rightarrow \ \underbrace{\mathbf{r}^{\mathrm{T}}\mathbf{x}}_{\delta} \geq 0$$

(Ranking) $(\exists \epsilon > 0)(\forall \tau \in \mathcal{T})$

$$L \models \quad \tau(\mathbf{x}\mathbf{x}') \geq 0 \ \rightarrow \ \mathbf{r}^{\mathrm{T}}\mathbf{x} - \mathbf{r}^{\mathrm{T}}\mathbf{x}' \geq \epsilon$$

Lexicographic Linear Ranking Function

Consider linear loop $L : \langle \Theta, \mathcal{T} \rangle$.

$$\langle \underbrace{\mathbf{r_1}^{\mathrm{T}}\mathbf{x}}_{\delta_1}, \ldots, \underbrace{\mathbf{r_k}^{\mathrm{T}}\mathbf{x}}_{\delta_k} \rangle \quad \text{and} \quad \pi : \mathcal{T} \to \{1, \ldots, k\}$$

for unknown coefficients $\mathbf{r}_{ij}$ and unknown π,
is a k**-component lexicographic linear ranking function** if

(Bounded) $(\forall \tau \in \mathcal{T})$

$$L \models \quad \tau(\mathbf{x}\mathbf{x}') \geq 0 \ \rightarrow \ \mathbf{r}^{\mathrm{T}}_{\pi(\tau)}\mathbf{x} \geq 0$$

(Ranking) $(\exists \epsilon > 0)(\forall \tau \in \mathcal{T})$

$$L \models \quad \tau(\mathbf{x}\mathbf{x}') \geq 0 \ \rightarrow \ \mathbf{r}^{\mathrm{T}}_{\pi(\tau)}\mathbf{x} - \mathbf{r}^{\mathrm{T}}_{\pi(\tau)}\mathbf{x}' \geq \epsilon$$

(Nonincreasing) $(\forall \tau \in \mathcal{T})$

$$L \models \quad (\forall j < \pi(\tau))[\tau(\mathbf{x}\mathbf{x}') \geq 0 \ \rightarrow \ \mathbf{r}^{\mathrm{T}}_{j}\mathbf{x} - \mathbf{r}^{\mathrm{T}}_{j}\mathbf{x}' \geq 0]$$

Linear Supporting Invariant

Ranking functions often require **supporting invariants**.

Consider linear loop $L : \langle \Theta, \mathcal{T} \rangle$.
For $\mathbf{Ix} \geq \mathbf{0}$ a template ($\mathbf{I}_{ij}$ are unknown coefficients),
$\mathbf{Ix} \geq 0$ is an ℓ**-conjunct linear invariant** if

(Initiation)

$$\Theta \mathbf{x} \geq 0 \;\rightarrow\; \mathbf{Ix} \geq \mathbf{0}$$

(Consecution) $(\forall \tau \in \mathcal{T})$

$$\mathbf{Ix} \geq 0 \;\wedge\; \tau(\mathbf{xx}') \geq 0 \;\rightarrow\; \mathbf{Ix}' \geq \mathbf{0}$$

Inductive assertion $\Rightarrow$ invariant

How do we find the unknown coefficients in the template?

Farkas Lemma (1894)

System of linear inequalities over real variables $\mathbf{x} = \{x_1, \ldots, x_n\}$:

$$S: \left[\begin{array}{ccccccccc} a_{1,1}x_1 & + & \cdots & + & a_{1,n}x_n & + & b_1 & \geq & 0 \\ \vdots & & & & \vdots & & \vdots & & \\ a_{m,1}x_1 & + & \cdots & + & a_{m,n}x_n & + & b_m & \geq & 0 \end{array}\right]$$

S entails linear inequality

$$\psi: \; c_1x_1 + \cdots + c_nx_n + d \geq 0$$

if and only if there exist real numbers $\lambda_1, \ldots, \lambda_m \geq 0$ such that

$$c_1 = \sum_{i=1}^{m} \lambda_i a_{i,1} \quad \cdots \quad c_n = \sum_{i=1}^{m} \lambda_i a_{i,n} \quad d \geq \left(\sum_{i=1}^{m} \lambda_i b_i\right)$$

Synthesis Overview

Consider loop $L : \langle \Theta, \mathcal{T} \rangle$ over $\mathcal{V} = \{x_1, x_2, \ldots, x_m\}$.

Templates:

- **expression** $\mathbf{c}^{\mathrm{T}}\mathbf{x}$, unknown coefficient vector $\mathbf{c}$
- **assertion** $\mathbf{Cx} \geq 0$, unknown coefficient matrix $\mathbf{C}$

Given templates

- ℓ-conjunct invariant template $\boxed{\mathbf{Ix} \geq \mathbf{0}}$ ($\mathbf{I}$ has ℓ rows)
- k-component lexicographic ranking function templates

$$\boxed{\{\mathbf{c_1}^{\mathrm{T}}\mathbf{x},\ \ldots, \mathbf{c_k}^{\mathrm{T}}\mathbf{x}\}}$$

Apply Farkas Lemma **rules** to encode **ranking function** and **supporting invariant** conditions.

Either use **given** π or **synthesize** it.

Farkas Lemma Rules: Invariant

(Initiation)

$$\mathbb{I}: \quad \frac{\Theta\mathbf{x} \ \geq \mathbf{0}}{\mathbf{I}\mathbf{x} \ \geq \mathbf{0}}$$

(Consecution)

$$\mathbb{C}_i: \quad \frac{\begin{array}{ll}\mathbf{I}\mathbf{x} & \geq \mathbf{0} \\ \tau_i(\mathbf{x}\mathbf{x}') & \geq \mathbf{0}\end{array}}{\mathbf{I}\mathbf{x}' \ \geq \mathbf{0}}$$

(Disabled)

$$\mathbb{D}_i: \quad \frac{\begin{array}{ll}\mathbf{I}\mathbf{x} & \geq \mathbf{0} \\ \tau_i(\mathbf{x}\mathbf{x}') & \geq \mathbf{0}\end{array}}{-1 \ \geq \mathbf{0}} \quad \leftarrow \text{false}$$

Farkas Lemma Rules: Ranking Function

(Bounded)

$$\begin{array}{rll} & \mathbf{Ix} & \geq \mathbf{0} \\ \mathbb{B}_i: & \tau_i(\mathbf{x}\mathbf{x}') & \geq \mathbf{0} \\ \hline & \mathbf{c}^{\mathrm{T}}\mathbf{x} & \geq \mathbf{0} \end{array}$$

(Ranking)

$$\begin{array}{rll} & \mathbf{Ix} & \geq \mathbf{0} \\ \mathbb{R}_i: & \tau_i(\mathbf{x}\mathbf{x}') & \geq \mathbf{0} \\ \hline & \mathbf{c}^{\mathrm{T}}\mathbf{x} - \mathbf{c}^{\mathrm{T}}\mathbf{x}' - \epsilon & \geq \mathbf{0} \end{array}$$

Example: $\mathbb{R}_1$ for GCD

$\tau_1 : \{y_1 \geq y_2 + 1\} \Rightarrow \{y_1' = y_1 - y_2,\ y_2' = y_2\}$

λ_1	$i_{1,1}y_1$	$+$	$i_{1,2}y_2$					$+$	$i_{1,3}$	$\geq$	0
λ_2	$i_{2,1}y_1$	$+$	$i_{2,2}y_2$					$+$	$i_{2,3}$	$\geq$	0
λ_3	y_1	$-$	y_2					$-$	1	$\geq$	0
λ_4	y_1	$-$	y_2	$-$	y_1'					$=$	0
λ_5			$-y_2$			$+$	y_2'			$=$	0
	c_1y_1	$+$	c_2y_2	$-$	c_1y_1'	$-$	c_2y_2'	$-$	ϵ	$\geq$	0

$$\Downarrow$$

$$\begin{aligned} \lambda_1 i_{1,1} + \lambda_2 i_{2,1} + \lambda_3 + \lambda_4 &= c_1 \\ \lambda_1 i_{1,2} + \lambda_2 i_{2,2} - \lambda_3 - \lambda_4 - \lambda_5 &= c_2 \\ -\lambda_4 &= -c_1 \\ \lambda_5 &= -c_2 \end{aligned} \qquad \begin{aligned} \lambda_1 i_{1,3} + \lambda_2 i_{2,3} - \lambda_3 &\leq -\epsilon \\ \lambda_1, \lambda_2, \lambda_3 &\geq 0 \\ \epsilon &> 0 \end{aligned}$$

Constraints are over $\{c_1, c_2, \epsilon, \lambda_1, \ldots, \lambda_5, i_{1,1}, \ldots, i_{2,3}\}$.

Generated Constraints

- Constraints are **linear** if no invariant template is given.
- Constraints are **parametric linear** otherwise.
  - Linear, except for a few bilinear quadratic terms
    (a λ and a supporting invariant template coefficient,
    e.g., $\lambda_1 i_{1,1}$)
  - Decidable [**Tarski 1951**]
  - Generic solvers based on CAD [**Collins 1975**]
    (**e.g.**, Mathematica, Redlog)
  - Specialized solvers
    [**Sankaranarayanan** *et al.* **2004**], [**Bradley** *et al.* **2005**]

Synthesis: Soundness and Completeness

Special Case

Linear loop $L : \langle \Theta, \mathcal{T} \rangle$ **has a**
linear ranking function
supported by an ℓ-conjunct linear invariant

$$\Leftrightarrow$$

the constraint system generated by

$$\mathbb{I} \ \wedge \ \bigwedge_{\tau_i \in \mathcal{T}} (\mathbb{D}_i \vee (\mathbb{C}_i \wedge \mathbb{B}_i \wedge \mathbb{R}_i))$$

is **satisfiable**.

Example: GCD

$$
\begin{aligned}
\Theta: &\quad \{y_1 \geq 1,\ y_2 \geq 1\} \\
\tau_1: &\quad \{y_1 \geq y_2 + 1\} \Rightarrow \{y_1' = y_1 - y_2,\ y_2' = y_2\} \\
\tau_2: &\quad \{y_2 \geq y_1 + 1\} \Rightarrow \{y_2' = y_2 - y_1,\ y_1' = y_1\}
\end{aligned}
$$

Find

- a linear ranking function $\mathbf{c}^{\mathrm{T}}\mathbf{x}$

$$c_1 y_1 + c_2 y_2 + c_3$$

- with a 2-conjunct supporting invariant $\mathbf{Ix} \geq \mathbf{0}$

$$
\begin{bmatrix} i_{1,1} & i_{1,2} & i_{1,3} \\ i_{2,1} & i_{2,2} & i_{2,3} \end{bmatrix}
\begin{bmatrix} y_1 \\ y_2 \\ 1 \end{bmatrix} \geq \mathbf{0}
$$

Example: GCD Invariant

Initiation

$$\mathbb{I}: \quad \begin{array}{l} y_1 \geq 1 \\ y_2 \geq 1 \\ \hline i_{1,1}y_1 + i_{1,2}y_2 + i_{1,3} \geq 0 \\ i_{2,1}y_1 + i_{2,2}y_2 + i_{2,3} \geq 0 \end{array}$$

Consecution

$$\mathbb{C}_1: \quad \begin{array}{l} i_{1,1}y_1 + i_{1,2}y_2 + i_{1,3} \geq 0 \\ i_{2,1}y_1 + i_{2,2}y_2 + i_{2,3} \geq 0 \\ y_1 \geq y_2 + 1 \\ y_1' = y_1 - y_2 \\ y_2' = y_2 \\ \hline i_{1,1}y_1' + i_{1,2}y_2' + i_{1,3} \geq 0 \\ i_{2,1}y_1' + i_{2,2}y_2' + i_{2,3} \geq 0 \end{array}$$

$$\mathbb{C}_2: \quad \begin{array}{l} i_{1,1}y_1 + i_{1,2}y_2 + i_{1,3} \geq 0 \\ i_{2,1}y_1 + i_{2,2}y_2 + i_{2,3} \geq 0 \\ y_2 \geq y_1 + 1 \\ y_2' = y_2 - y_1 \\ y_1' = y_1 \\ \hline i_{1,1}y_1' + i_{1,2}y_2' + i_{1,3} \geq 0 \\ i_{2,1}y_1' + i_{2,2}y_2' + i_{2,3} \geq 0 \end{array}$$

Example: GCD Ranking Function

Bounded

$$\mathbb{B}_1: \quad \frac{\begin{array}{l} i_{1,1}y_1 + i_{1,2}y_2 + i_{1,3} \geq 0 \\ i_{2,1}y_1 + i_{2,2}y_2 + i_{2,3} \geq 0 \\ y_1 \geq y_2 + 1 \end{array}}{c_1y_1 + c_2y_2 + c_3 \geq 0} \qquad \mathbb{B}_2: \quad \frac{\begin{array}{l} i_{1,1}y_1 + i_{1,2}y_2 + i_{1,3} \geq 0 \\ i_{2,1}y_1 + i_{2,2}y_2 + i_{2,3} \geq 0 \\ y_2 \geq y_1 + 1 \end{array}}{c_1y_1 + c_2y_2 + c_3 \geq 0}$$

Ranking

$$\mathbb{R}_1: \quad \frac{\begin{array}{l} i_{1,1}y_1 + i_{1,2}y_2 + i_{1,3} \geq 0 \\ i_{2,1}y_1 + i_{2,2}y_2 + i_{2,3} \geq 0 \\ y_1 \geq y_2 + 1 \\ y_1' = y_1 - y_2 \\ y_2' = y_2 \end{array}}{c_1y_1 + c_2y_2 \geq c_1y_1' + c_2y_2' + \epsilon} \qquad \mathbb{R}_2: \quad \frac{\begin{array}{l} i_{1,1}y_1 + i_{1,2}y_2 + i_{1,3} \geq 0 \\ i_{2,1}y_1 + i_{2,2}y_2 + i_{2,3} \geq 0 \\ y_2 \geq y_1 + 1 \\ y_2' = y_2 - y_1 \\ y_1' = y_1 \end{array}}{c_1y_1 + c_2y_2 \geq c_1y_1' + c_2y_2' + \epsilon}$$

Example: GCD Synthesis

$$
\begin{aligned}
\Theta: &\quad \{y_1 \geq 1,\ y_2 \geq 1\} \\
\tau_1: &\quad \{y_1 \geq y_2 + 1\} \Rightarrow \{y_1' = y_1 - y_2,\ y_2' = y_2\} \\
\tau_2: &\quad \{y_2 \geq y_1 + 1\} \Rightarrow \{y_2' = y_2 - y_1,\ y_1' = y_1\}
\end{aligned}
$$

Solving the constraint system induced by

$$\mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{B}_1 \wedge \mathbb{B}_2 \wedge \mathbb{R}_1 \wedge \mathbb{R}_2$$

reveals ranking function

$$c_1 = c_2 = 1,\ c_3 = 0 \ \Rightarrow\ \boxed{y_1 + y_2}$$

with $\epsilon = 1$, supported by the invariants

$$
\begin{aligned}
i_{1,1} = 1,\ i_{1,2} = 0,\ i_{1,3} = -1 \ &\Rightarrow \\
i_{2,1} = 0,\ i_{2,2} = 1,\ i_{2,3} = -1 \ &\Rightarrow
\end{aligned}
\qquad
\boxed{\begin{array}{c} y_1 \geq 1 \\ y_2 \geq 1 \end{array}}
$$

which proves that GCD always terminates.

Synthesis of Linear Lexicographic Ranking Functions with Supporting Invariants

Farkas Lemma Rules: Invariant

(Initiation)

$$\mathbb{I}: \quad \frac{\Theta\mathbf{x} \ \geq \mathbf{0}}{\mathbf{I}\mathbf{x} \ \geq \mathbf{0}}$$

(Consecution)

$$\mathbb{C}_i: \quad \frac{\begin{array}{ll}\mathbf{I}\mathbf{x} & \geq \mathbf{0} \\ \tau_i(\mathbf{x}\mathbf{x}') & \geq \mathbf{0}\end{array}}{\mathbf{I}\mathbf{x}' \ \geq \mathbf{0}}$$

(Disabled)

$$\mathbb{D}_i: \quad \frac{\begin{array}{ll}\mathbf{I}\mathbf{x} & \geq \mathbf{0} \\ \tau_i(\mathbf{x}\mathbf{x}') & \geq \mathbf{0}\end{array}}{-1 \ \geq \mathbf{0}} \quad \leftarrow \text{false}$$

Farkas Lemma Rules: Ranking Function

(Bounded)

$$\mathbb{B}_{ij}: \quad \begin{array}{ll} \mathbf{Ix} & \geq \mathbf{0} \\ \tau_i(\mathbf{xx'}) & \geq \mathbf{0} \\ \hline \mathbf{c_j}^{\mathrm{T}}\mathbf{x} & \geq \mathbf{0} \end{array}$$

(Ranking)

$$\mathbb{R}_{ij}: \quad \begin{array}{ll} \mathbf{Ix} & \geq \mathbf{0} \\ \tau_i(\mathbf{xx'}) & \geq \mathbf{0} \\ \hline \mathbf{c_j}^{\mathrm{T}}\mathbf{x} - \mathbf{c_j}^{\mathrm{T}}\mathbf{x'} - \epsilon & \geq \mathbf{0} \end{array}$$

(Nonincreasing)

$$\mathbb{N}_{ij}: \quad \begin{array}{ll} \mathbf{Ix} & \geq \mathbf{0} \\ \tau_i(\mathbf{xx'}) & \geq \mathbf{0} \\ \hline \mathbf{c_j}^{\mathrm{T}}\mathbf{x} - \mathbf{c_j}^{\mathrm{T}}\mathbf{x'} & \geq \mathbf{0} \end{array}$$

Synthesis: Soundness and Completeness

Theorem (General Case)

Linear loop $L : \langle \Theta, \mathcal{T} \rangle$ has a
k-lexicographic linear ranking function
supported by an ℓ-conjunct linear invariant

$$\Leftrightarrow$$

the constraint system generated by

$$\mathbb{I} \ \wedge \ \bigwedge_{\tau_i \in \mathcal{T}} \big(\mathbb{D}_i \vee \big(\mathbb{C}_i \wedge \mathbb{B}_{i,\pi(i)} \wedge \mathbb{R}_{i,\pi(i)} \big) \big) \ \wedge \ \bigwedge_{\tau_i \in \mathcal{T},\ j < \pi(i)} \big(\mathbb{D}_i \vee \mathbb{N}_{ij} \big)$$

is **satisfiable** for some $\pi : \mathcal{T} \to \{1, \ldots, k\}$.

Synthesizing Lexicographic Ranking Functions

The theorem asks for a map π. Can we efficiently find the map?

Overview:

- Initially propose one template component per transition

  $$\{\mathbf{c_1}^{\mathrm{T}}\mathbf{x}, \ldots, \mathbf{c_n}^{\mathrm{T}}\mathbf{x}\}$$

  Inexpensive because template coefficients $\mathbf{c_i}$ appear only **linearly** in generated constraints.

- Incrementally build linear order $\prec$ over $\mathcal{T}$.
  Linear order gives π.

- Use intermediate partial orders to generate and solve constraint systems to guide search.

Simplified Farkas Lemma Rules

One component per transition.

(Bounded)

$$\mathbb{B}_i : \begin{array}{ll} \mathbf{Ix} & \geq \mathbf{0} \\ \tau_i(\mathbf{xx'}) & \geq \mathbf{0} \\ \hline \mathbf{c_i}^{\mathrm{T}}\mathbf{x} & \geq \mathbf{0} \end{array}$$

(Ranking)

$$\mathbb{R}_i : \begin{array}{ll} \mathbf{Ix} & \geq \mathbf{0} \\ \tau_i(\mathbf{xx'}) & \geq \mathbf{0} \\ \hline \mathbf{c_i}^{\mathrm{T}}\mathbf{x} - \mathbf{c_i}^{\mathrm{T}}\mathbf{x'} - \epsilon & \geq \mathbf{0} \end{array}$$

Synthesizing Lexicographic Ranking Functions

Incrementally build $\prec$ over $\mathcal{T}$:

1. Guess $\tau_i \prec \tau_j$:

$$\langle \ldots, \mathbf{c_i}^{\mathrm{T}}\mathbf{x}, \ldots, \mathbf{c_j}^{\mathrm{T}}\mathbf{x}, \ldots \rangle$$

2. Generate constraint system and solve.
   - $\mathbb{I}$, $\mathbb{C}_i$, $\mathbb{D}_i$, $\mathbb{B}_i$, $\mathbb{R}_i$ are fixed.
   - $\tau_i \prec \tau_j$ induces $\mathbb{N}_{ji}$: τ_j should not increase $\mathbf{c_i}^{\mathrm{T}}\mathbf{x}$

   Satisfiable $\Rightarrow$ Continue search

   Unsatisfiable $\Rightarrow$ Try $\tau_i \succ \tau_j$

   Satisfiable $\Rightarrow$ Continue search

   Unsatisfiable $\Rightarrow$ Backtrack

3. Finished when order is **linear**
   and generated constraint system is **satisfiable**.

One Constraint System per Component

Given lexicographic template components

$$\{\mathbf{c_1}^{\mathrm{T}}\mathbf{x}, \ldots, \mathbf{c_n}^{\mathrm{T}}\mathbf{x}\}$$

and **partial order** $\prec$ over $\mathcal{T}$, solve n constraint systems induced by

$$\underbrace{\mathbb{I} \wedge \bigwedge_{\tau_i} (\mathbb{D}_i \vee \mathbb{C}_i)}_{\text{Invariant}} \quad \wedge \quad \underbrace{(\mathbb{D}_j \vee (\mathbb{B}_j \wedge \mathbb{R}_j))}_{\tau_j \text{ ranked by } \mathbf{c_j}^{\mathrm{T}}}$$

$$\wedge \quad \underbrace{\bigwedge_{\tau_i \text{ s.t. } \tau_j \prec \tau_i} (\mathbb{D}_i \vee \mathbb{N}_{ij})}_{\tau_i \text{ does not increase } \mathbf{c_j}^{\mathrm{T}} \text{ if } \tau_j \prec \tau_i}$$

$$\langle \ldots, \quad \underset{\uparrow}{\mathbf{c_j}^{\mathrm{T}}\mathbf{x}}, \quad \ldots, \quad \underset{\uparrow}{\mathbf{c_i}^{\mathrm{T}}\mathbf{x}}, \quad \ldots \rangle$$

$$\tau_j \qquad \tau_i$$

for $j \in \{1, \ldots, n\}$.

One Constraint System per Component

Advantages:

- Multiple smaller constraint systems are easier to solve in practice.
- Invariant template may be instantiated differently for each constraint system.

Example: McCarthy 91

$$
\begin{array}{lll}
\Theta: & \{s = 1\} & \\
\tau_1: & \{x \geq 101,\ s \leq 0\} \Rightarrow \{x' = x - 10,\ s' = s - 1\} & \left.\begin{array}{l}\\\\\end{array}\right\} \text{split of} \\
\tau_2: & \{x \geq 101,\ s \geq 2\} \Rightarrow \{x' = x - 10,\ s' = s - 1\} & \left.\begin{array}{l}\\\\\end{array}\right\} s \neq 1 \\
\tau_3: & \{x \leq 100\} \Rightarrow \{x' = x + 11,\ s' = s + 1\} &
\end{array}
$$

Find

- a 3-component lexicographic linear ranking function

$$\langle \mathbf{r_1}^{\mathrm{T}}\mathbf{x},\ \mathbf{r_2}^{\mathrm{T}}\mathbf{x},\ \mathbf{r_3}^{\mathrm{T}}\mathbf{x} \rangle$$

$$\pi : \mathcal{T} \to \{1, 2, 3\}$$

- with a 1-conjunct supporting invariant $\mathbf{Ix} \geq 0$

Example: McCarthy 91

Iteration 1:

- One template per transition.

$$\{\mathbf{c_1}^{\mathrm{T}}\mathbf{x},\ \mathbf{c_2}^{\mathrm{T}}\mathbf{x},\ \mathbf{c_3}^{\mathrm{T}}\mathbf{x}\}$$

- No order assumed between transitions.
- Three sets of conditions:

$$\begin{aligned}
\mathbf{c_1}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_1 \wedge \mathbb{R}_1 \\
\mathbf{c_2}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_2 \wedge \mathbb{R}_2 \\
\mathbf{c_3}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_3 \wedge \mathbb{R}_3
\end{aligned}$$

- Induced constraint systems are **satisfiable**.

Example: McCarthy 91

Constraints on $\mathbf{c_1}^{\mathrm{T}}\mathbf{x}$:

Initiation

$$\mathbb{I}: \quad \frac{s = 1}{i_1 s + i_2 x + i_3 \geq 0}$$

Consecution

$$\mathbb{C}_1: \quad \frac{\begin{array}{l} i_1 s + i_2 x + i_3 \geq 0 \\ x \geq 101 \\ s \leq 0 \\ x' = x - 10 \\ s' = s - 1 \end{array}}{i_1 s' + i_2 x' + i_3 \geq 0}$$

$$\mathbb{C}_2: \quad \frac{\begin{array}{l} i_1 s + i_2 x + i_3 \geq 0 \\ x \geq 101 \\ s \geq 2 \\ x' = x - 10 \\ s' = s - 1 \end{array}}{i_1 s' + i_2 x' + i_3 \geq 0}$$

$$\mathbb{C}_3: \quad \frac{\begin{array}{l} i_1 s + i_2 x + i_3 \geq 0 \\ x \leq 100 \\ x' = x + 11 \\ s' = s + 1 \end{array}}{i_1 s' + i_2 x' + i_3 \geq 0}$$

Example: McCarthy 91

Bounded

$$\mathbb{B}_1 : \quad \frac{\begin{array}{l} i_1 s + i_2 x + i_3 \geq 0 \\ x \geq 101 \\ s \leq 0 \end{array}}{c_{1,1} s + c_{1,2} x + c_{1,3} \geq 0}$$

Ranking

$$\mathbb{R}_1 : \quad \frac{\begin{array}{l} i_1 s + i_2 x + i_3 \geq 0 \\ x \geq 101 \\ s \leq 0 \\ x' = x - 10 \\ s' = s - 1 \end{array}}{c_{1,1} s + c_{1,2} x \geq c_{1,1} s' + c_{1,2} x' + \epsilon}$$

Example: McCarthy 91

Iteration 2:

- Guess $\tau_3 \prec \tau_2$.

$$\langle \ldots, \mathbf{c_3}^{\mathrm{T}}\mathbf{x}, \ldots, \mathbf{c_2}^{\mathrm{T}}\mathbf{x}, \ldots \rangle$$

  τ_2 should not increase $\mathbf{c_3}^{\mathrm{T}}\mathbf{x}$.

- Three sets of conditions:

$$\begin{aligned}
\mathbf{c_1}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_1 \wedge \mathbb{R}_1 \\
\mathbf{c_2}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_2 \wedge \mathbb{R}_2 \\
\mathbf{c_3}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_3 \wedge \mathbb{R}_3 \wedge \mathbb{N}_{2,3}
\end{aligned}$$

- Induced constraint systems are **satisfiable**.

Example: McCarthy 91

Iteration 3:

- Guess $\tau_1 \prec \tau_3$.

$$\langle \ldots, \mathbf{c_1}^{\mathrm{T}}\mathbf{x}, \ldots, \mathbf{c_3}^{\mathrm{T}}\mathbf{x}, \ldots \rangle$$

  τ_3 should not increase $\mathbf{c_1}^{\mathrm{T}}\mathbf{x}$.

- Three sets of conditions:

$$\begin{aligned}
\mathbf{c_1}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_1 \wedge \mathbb{R}_1 \wedge \mathbb{N}_{3,1} \\
\mathbf{c_2}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_2 \wedge \mathbb{R}_2 \\
\mathbf{c_3}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_3 \wedge \mathbb{R}_3 \wedge \mathbb{N}_{2,3}
\end{aligned}$$

- Induced constraint system for $\mathbf{c_1}^{\mathrm{T}}\mathbf{x}$ is **unsatisfiable**.

Example: McCarthy 91

Iteration 3:

- Try $\tau_3 \prec \tau_1$ instead.

$$\langle \ldots, \mathbf{c_3}^{\mathrm{T}}\mathbf{x}, \ldots, \mathbf{c_1}^{\mathrm{T}}\mathbf{x}, \ldots \rangle$$

  τ_1 should not increase $\mathbf{c_3}^{\mathrm{T}}\mathbf{x}$.

- Three sets of conditions:

$$\begin{aligned}
\mathbf{c_1}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_1 \wedge \mathbb{R}_1 \\
\mathbf{c_2}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_2 \wedge \mathbb{R}_2 \\
\mathbf{c_3}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_3 \wedge \mathbb{R}_3 \wedge \mathbb{N}_{2,3} \wedge \mathbb{N}_{1,3}
\end{aligned}$$

- Induced constraint systems are **satisfiable**.

Example: McCarthy 91

Iteration 4:

- Guess $\tau_2 \prec \tau_1$ (but $\tau_1 \prec \tau_2$ works, too).

$$\langle \ldots, \mathbf{c_2}^{\mathrm{T}}\mathbf{x}, \ldots, \mathbf{c_1}^{\mathrm{T}}\mathbf{x}, \ldots \rangle$$

  τ_1 should not increase $\mathbf{c_2}^{\mathrm{T}}\mathbf{x}$.

- Three sets of conditions:

$$\begin{aligned}
\mathbf{c_1}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_1 \wedge \mathbb{R}_1 \\
\mathbf{c_2}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_2 \wedge \mathbb{R}_2 \wedge \mathbb{N}_{1,2} \\
\mathbf{c_3}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_3 \wedge \mathbb{R}_3 \wedge \mathbb{N}_{2,3} \wedge \mathbb{N}_{1,3}
\end{aligned}$$

- Induced constraint systems are **satisfiable**.

Example: McCarthy 91

Iteration 4:

- $\prec$ is a linear order:

$$\langle \mathbf{c_3}^{\mathrm{T}}\mathbf{x},\ \mathbf{c_2}^{\mathrm{T}}\mathbf{x},\ \mathbf{c_1}^{\mathrm{T}}\mathbf{x} \rangle$$

$$\tau_3 \prec \tau_2 \prec \tau_1$$

- $\prec$ gives π:

$$\pi(\tau_1) = 3,\ \pi(\tau_2) = 2,\ \pi(\tau_3) = 1$$

Example: McCarthy 91

$$
\begin{aligned}
\Theta: &\quad \{s = 1\} \\
\tau_1: &\quad \{x \geq 101,\ s \leq 0\} \Rightarrow \{x' = x - 10,\ s' = s - 1\} \\
\tau_2: &\quad \{x \geq 101,\ s \geq 2\} \Rightarrow \{x' = x - 10,\ s' = s - 1\} \\
\tau_3: &\quad \{x \leq 100\} \Rightarrow \{x' = x + 11,\ s' = s + 1\}
\end{aligned}
$$

Solving the final constraint systems reveals ranking function

$$\langle 10s - x + 90,\ x,\ x \rangle$$

$$\pi(\tau_1) = 3,\ \pi(\tau_2) = 2,\ \pi(\tau_3) = 1$$

supported by the invariant

$$s \geq 1$$

which proves that McCarthy 91 always terminates.

In Practice

Name	LOC	L	A	P	P/A	P/L	Sec
`meschach`	28K	911	778	758	97%	83%	64
`gnuplot`	50K	826	312	301	96%	36%	88
`gaim`	57K	594	54	52	96%	8%	94
`ffmpeg`	108K	2674	2115	2081	98%	78%	198

- Prototype **unsound** abstraction of C loops, using CIL.
  Why unsound?
  - Overflows, `unsigned`, `doubles` abstracted as $\mathbb{R}$s, *etc.*
  - Aliasing, globals changed by function calls, *etc.*

  In principle, can be sound — an engineering task.
- Synthesis of lexicographic linear ranking functions.
  Lexicographic function needed for 10 loops.

Example of Failed Abstraction

```
List * iter = items;
while (iter != NULL) {
  ...
  iter = iter->next;
}
```

Proving termination requires:

- proving that `items` is noncircular (**nontrivial**);
- abstracting iteration to counting down (**trivial**).

Example of Failed Abstraction

```
char * ptr = input;
while (*ptr != '\0') {
  ...
  ptr++;
}
```

Proving termination requires:

- proving that `input` is a well-formed C string (**nontrivial**).
- abstracting pointer arithmetic to counting down (**trivial**).

Reasons for Failed Proofs

1. **Prototype** abstracter!
2. Need for invariants. Examples:

   `i = 2 * i;` $i \geq 0$ to deduce increase in `i`

   `i = i + k;` $k > 0$ to deduce increase in `i`
3. Need for **summarizing** embedded loops. Example:

```
while (i < n) {
  while (...) { i++; }    ← summarize with i' ≥ i
  i++;
}
```

4. Need for function invariants. Example:

   `i = i + strlen(str);` knowledge about `str` and `strlen`
5. Loop does not terminate.

Appendix

Expanding a Farkas Lemma Rule

$$
\begin{array}{lll}
 & \mathbf{Ix} & \geq \mathbf{0} \\
\mathbb{R}_{ij}: & \tau_i(\mathbf{xx'}) & \geq \mathbf{0} \\
\hline
 & \mathbf{c_j}^{\mathrm{T}}\mathbf{x} - \mathbf{c_j}^{\mathrm{T}}\mathbf{x'} - \epsilon & \geq \mathbf{0}
\end{array}
$$

$\mathbf{x} = (x_1, \ldots, x_m, 1)^{\mathrm{T}}$

$(\mathbf{xx'}) = (x_1, \ldots, x_m, x'_1, \ldots, x'_m, 1)^{\mathrm{T}}$

$\Downarrow$

$$
\begin{array}{lrl}
\lambda_I & \mathbf{Ix} + \mathbf{i} & \geq \mathbf{0} \\
\lambda_G & \mathbf{G_i x} + \mathbf{g_i} & \geq \mathbf{0} \\
\lambda_U & \mathbf{U_i x} + \mathbf{V_i x'} + \mathbf{u_i} & \geq \mathbf{0} \\
\hline
 & \mathbf{c_j}^{\mathrm{T}}\mathbf{x} - \mathbf{c_j}^{\mathrm{T}}\mathbf{x'} - \epsilon & \geq \mathbf{0}
\end{array}
$$

Expand assertions:

$\mathbf{x} = (x_1, \ldots, x_m)^{\mathrm{T}}$

$\mathbf{x'} = (x'_1, \ldots, x'_m)^{\mathrm{T}}$

$\mathbf{I}, \mathbf{i}$ define Θ

$\mathbf{G_i}, \mathbf{g_i}, \mathbf{U_i}, \mathbf{V_i}, \mathbf{u_i}$ define τ_i

$\Downarrow$

$$
\begin{array}{rcl}
\lambda_I^{\mathrm{T}}\mathbf{I} + \lambda_G^{\mathrm{T}}\mathbf{G_i} + \lambda_U^{\mathrm{T}}\mathbf{U_i} & = & \mathbf{c_j} \\
\lambda_U^{\mathrm{T}}\mathbf{V_i} & = & -\mathbf{c_j} \\
\lambda_I^{\mathrm{T}}\mathbf{i} + \lambda_G^{\mathrm{T}}\mathbf{g_i} + \lambda_U^{\mathrm{T}}\mathbf{u_i} & \leq & -\epsilon \\
\lambda_I, \lambda_G, \lambda_U & \geq & 0 \\
\epsilon & > & 0
\end{array}
$$

Constraints over $\{\lambda_I, \lambda_G, \lambda_U, \mathbf{c_j}, \epsilon\}$

Lexicographic Synthesis

```
let lex L : ⟨Θ, 𝒯⟩ =
  let sat o = (∀τ ∈ 𝒯)(satisfiable L τ o) in
  let rec search o =
    sat o and
      match choose o with
      |          None  →  true
                            (search (o ∪ (τ ≺ τ')))
      | Some (τ, τ')  →                  or
                            (search (o ∪ (τ' ≺ τ)))
  in
  search {}
```

Lexicographic Synthesis

Calling

$$\mathit{satisfiable}\ L\ \tau_j\ o$$

generates and solves a constraint system for L, τ, and o, in which

$$\mathbb{N}_{ij}$$

is imposed iff

$$(\tau_j \prec \tau_i) \in o$$

Solving **partial** constraint systems **guides** the construction of lexicographic ranking functions.